May, 2013

Dynamic Provisioning in Next-Generation Data Centers with On-site Power Production

Jinlong Tu
Lian Lu
Minghua Chen
Ramesh Sitaraman, *University of Massachusetts - Amherst*

Dynamic Provisioning in Next-Generation Data Centers with On-site Power Production

Jinlong Tu, Lian Lu and Minghua Chen
Department of Information Engineering
The Chinese University of Hong Kong

Ramesh K. Sitaraman
Department of Computer Science
University of Massachusetts at Amherst
& Akamai Technologies

ABSTRACT

The critical need for clean and economical sources of energy is transforming data centers that are primarily energy consumers to also energy producers. We focus on minimizing the operating costs of next-generation data centers that can jointly optimize the energy supply from on-site generators and the power grid, and the energy demand from servers as well as power conditioning and cooling systems. We formulate the cost minimization problem and present an offline optimal algorithm. For "on-grid" data centers that use only the grid, we devise a deterministic online algorithm that achieves the best possible competitive ratio of $2-\alpha_s$, where α_s is a normalized look-ahead window size. The competitive ratio of an online algorithm is defined as the maximum ratio (over all possible inputs) between the algorithm's cost (with no or limited look-ahead) and the offline optimal assuming complete future information. We remark that the results hold as long as the overall energy demand (including server, cooling, and power conditioning) is a convex and increasing function in the total number of active servers and also in the total server load. For "hybrid" data centers that have on-site power generation in addition to the grid, we develop an online algorithm that achieves a competitive ratio of at most $\frac{P_{\max}(2-\alpha_s)}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}(1+\alpha_g)}\right]$, where α_s and α_g are normalized look-ahead window sizes, $P_{\max}$ is the maximum grid power price, and L, c_o, and c_m are parameters of an on-site generator.

Using extensive workload traces from Akamai with the corresponding grid power prices, we simulate our offline and online algorithms in a realistic setting. Our offline (resp., online) algorithm achieves a cost reduction of 25.8% (resp., 20.7%) for a hybrid data center and 12.3% (resp., 7.3%) for an on-grid data center. The cost reductions are quite significant and make a strong case for a joint optimization of energy supply and energy demand in a data center. A hybrid data center provides about 13% additional cost reduction over an on-grid data center representing the additional cost benefits that on-site power generation provides over using the grid alone.

e-Energy'13, May 21–24, 2013, Berkeley, California, USA.

Categories and Subject Descriptors

F.1.2 [**Modes of Computation**]: Online computation; G.1.6 [**Optimization**]: Nonlinear programming; I.1.2 [**Algorithms**]: Analysis of algorithms; I.2.8 [**Problem Solving, Control Methods, and Search**]: Scheduling

General Terms

Algorithms, Performance

Keywords

data centers; dynamic provisioning; on-site power production; online algorithm

1. INTRODUCTION

Internet-scale cloud services that deploy large distributed systems of servers around the world are revolutionizing all aspects of human activity. The rapid growth of such services has lead to a significant increase in server deployments in data centers around the world. Energy consumption of data centers account for roughly 1.5% of the global energy consumption and is increasing at an alarming rate of about 15% on an annual basis [21]. The surging global energy demand relative to its supply has caused the price of electricity to rise, even while other operating expenses of a data center such as network bandwidth have decreased precipitously. Consequently, the energy costs now represent a large fraction of the operating expenses of a data center today [9], and decreasing the energy expenses has become a central concern for data center operators.

The emergence of energy as a central consideration for enterprises that operate large server farms is drastically altering the traditional boundary between a data center and a power utility (c.f. Figure 1). Traditionally, a data center hosts servers but buys electricity from an utility company through the power grid. However, the criticality of the energy supply is leading data centers to broaden their role to also generate much of the required power on-site, decreasing their dependence on a third-party utility. While data centers have always had generators as a short-term backup for when the grid fails, on-site generators for sustained power supply is a newer trend. For instance, Apple recently announced that it will build a massive data center for its iCloud services with 60% of its energy coming from its on-site generators that use "clean energy" sources such as fuel

cells with biogas and solar panels [25]. As another example, eBay recently announced that it will add a 6 MW facility to its existing data center in Utah that will be largely powered by on-site fuel cell generators [17]. The trend for *hybrid* data centers that generate electricity on-site (c.f. Figure 1) with reduced reliance on the grid is driven by the confluence of several factors. This trend is also mirrored in the broader power industry where the centralized model for power generation with few large power plants is giving way to a more distributed generation model [11] where many smaller on-site generators produce power that is consumed locally over a "micro-grid".

A key factor favoring on-site generation is the potential for cheaper power than the grid, especially during peak hours. On-site generation also reduces transmission losses that in turn reduce the effective cost, because the power is generated close to where it is consumed. In addition, another factor favoring on-site generation is a requirement for many enterprises to use cleaner renewable energy sources, such as Apple's mandate to use 100% clean energy in its data centers [6]. Such a mandate is more easily achievable with the enterprise generating all or most of its power on-site, especially since recent advances such as the fuel cell technology of Bloom Energy [7] make on-site generation economical and feasible. Finally, the risk of service outages caused by the failure of the grid, as happened recently when thunderstorms brought down the grid causing a denial-of-service for Amazon's AWS service for several hours [18], has provided greater impetus for on-site power generation that can sustain the data center for extended periods without the grid.

Our work focuses on the key challenges that arise in the emerging hybrid model for a data center that is able to simultaneously optimize *both* the generation and consumption of energy (c.f. Figure 1). In the traditional scenario, the utility is responsible for energy provisioning (**EP**) that has the goal of supplying energy as economically as possible to meet the energy demand, albeit the utility has no detailed knowledge and no control over the server workloads within a data center that drive the consumption of power. Optimal energy provisioning by the utility in isolation is characterized by the unit commitment problem [31, 36] that has been studied over the past decades. The energy provisioning problem takes as input the demand for electricity from the consumers and determines which power generators should be used at what time to satisfy the demand in the most economical fashion. Further, in a traditional scenario, a data center is responsible for capacity provisioning (**CP**) that has the goal of managing its server capacity to serve the incoming workload from end users while reducing the total energy demand of servers, as well as power conditioning and various cooling systems, but without detailed knowledge or control over the power generation. For instance, dynamic provisioning of server capacity by turning off some servers during periods of low workload to reduce the energy demand has been studied in recent years [23, 28, 10, 27].

The convergence of power generation and consumption within a single data center entity and the increasing impact of energy costs requires a new integrated approach to both energy provisioning (**EP**) and capacity provisioning (**CP**). A key contribution of our work is formulating and developing algorithms that simultaneously manage on-site power generation, grid power consumption, and server capacity with the goal of minimizing the operating cost of the data center.

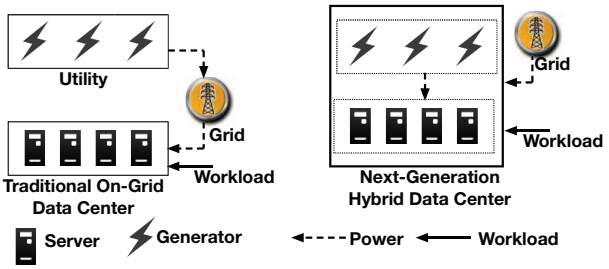


Figure 1: While an "on-grid" data center derives all its power from the grid, next-generation "hybrid" data centers have additional on-site power generation.

Online vs. Offline Algorithms. In designing algorithms for optimizing the operating cost of a hybrid data center, there are three time-varying inputs: the server workload $a(t)$ generated by service requests from users and the price of a unit energy from the grid $p(t)$, and the total power consumption function g_t for each time t where $1 \leq t \leq T$. We begin by investigating *offline* algorithms that minimize the operating cost with perfect knowledge of the entire input sequence $a(t)$, $p(t)$ and g_t, for $1 \leq t \leq T$. However, in real-life, the time-varying input sequences are not knowable in advance. In particular, the optimization must be performed in an *online* fashion where decisions at time t are made with the knowledge of inputs $a(\tau)$,$p(\tau)$ and g_τ, for $1 \leq \tau \leq t+w$, where $w \geq 0$ is a small (possibly zero) look-ahead window. Specifically, an online algorithm has no knowledge of inputs beyond the look-ahead window, *i.e.*, for time $t+w < \tau \leq T$. We assume the inputs within the look-ahead are perfectly known when analyzing the algorithm performance. In practice, short-term demand or grid price can be estimated rather accurately by various techniques including pattern analysis and time series analysis and prediction [19, 14]. As is typical in the study of online algorithms [12], we seek theoretical guarantees for our online algorithms by computing the *competitive ratio* that is ratio of the cost achieved by the online algorithm for an input to the optimal cost achieved for the same input by an offline algorithm. The competitive ratio is computed under a worst case scenario where an adversary picks the worst possible inputs for the online algorithm. Thus, a small competitive ratio provides a strong guarantee that the online algorithm will achieve a cost close to the offline optimal even for the worst case input.

Our Contributions. A key contribution of our work is to formulate and study data center cost minimization (**DCM**) that integrates energy procurement from the grid, energy production using on-site generators, and dynamic server capacity management. Our work jointly optimizes the two components of **DCM**: energy provisioning (**EP**) from the grid and generators and capacity provisioning (**CP**) of the servers.

- We theoretically evaluate the benefit of joint optimization by showing that optimizing energy provisioning (**EP**) and capacity provisioning (**CP**) separately results in a factor loss of optimality $\rho = LP_{\max}/(Lc_o + c_m)$ compared to optimizing them jointly, where $P_{\max}$ is the maximum grid power price, and L, c_o, and c_m are the capacity, incremental cost, and base cost of an on-site generator respectively. Further, we derive an efficient offline optimal algorithm for hybrid data centers that

Competitive Ratio	On-grid	Hybrid
No Look-ahead	2	$\frac{2P_{\max}}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}}\right]$
With Look-ahead	$2-\alpha_s$	$\frac{P_{\max}(2-\alpha_s)}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}(1+\alpha_g)}\right]$

Table 1: Summary of algorithmic results. The on-grid results are the best possible for any deterministic online algorithm.

jointly optimize **EP** and **CP** to minimize the data center's operating cost.

- For on-grid data centers, we devise an online deterministic algorithm that achieves a competitive ratio of $2-\alpha_s$, where $\alpha_s \in [0,1]$ is the normalized look-ahead window size. Further, we show that our algorithm has the best competitive ratio of any deterministic online algorithm for the problem (c.f. Table 1). For the more complex hybrid data centers, we devise an online deterministic algorithm that achieves a competitive ratio of $\frac{P_{\max}(2-\alpha_s)}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}(1+\alpha_g)}\right]$, where α_s and α_g are normalized look-ahead window sizes. Both online algorithms perform better as the look-ahead window increases, as they are better able to plan their current actions based on knowledge of future inputs. Interestingly, in the on-grid case, we show that there exists *fixed* threshold value for the look-ahead window for which the online algorithm matches the offline optimal in performance achieving a competitive ratio of 1, *i.e.*, there is no additional benefit gained by the online algorithm if its look-ahead is increased beyond the threshold.
- Using extensive workload traces from Akamai and the corresponding grid prices, we simulate our offline and online algorithms in a realistic setting with the goal of empirically evaluating their performance. Our offline optimal (resp., online) algorithm achieves a cost reduction of 25.8% (resp., 20.7%) for a hybrid data center and 12.3% (resp., 7.3%) for an on-grid data center. The cost reduction is computed in comparison with the baseline cost achieved by the current practice of statically provisioning the servers and using only the power grid. The cost reductions are quite significant and make a strong case for utilizing our joint cost optimization framework. Furthermore, our online algorithms obtain almost the same cost reduction as the offline optimal solution even with a small look-ahead of 6 hours, indicating the value of short-term prediction of inputs.
- A hybrid data center provides about 13% additional cost reduction over an on-grid data center representing the additional cost benefits that on-site power generation provides over using the grid alone. Interestingly, it is sufficient to deploy a partial on-site generation capacity that provides 60% of the peak power requirements of the data center to obtain over 95% of the additional cost reduction. This provides strong motivation for a traditional on-grid data center to deploy at least a partial on-site generation capability to save costs.

Due to space limitations, all proofs are in our technical report [39].

2. THE DATA CENTER COST MINIMIZATION PROBLEM

We consider the scenario where a data center can jointly optimize energy production, procurement, and consumption so as to minimize its operating expenses. We refer to this data center cost minimization problem as **DCM**. To study **DCM**, we model how energy is produced using on-site power generators, how it can be procured from the power grid, and how data center capacity can be provisioned dynamically in response to workload. While some of these aspects have been studied independently, our work is unique in optimizing these dimensions simultaneously as next-generation data centers can. Our algorithms minimize cost by use of techniques such as: (i) dynamic capacity provisioning of servers – turning off unnecessary servers when workload is low to reduce the energy consumption (ii) opportunistic energy procurement – opting between the on-site and grid energy sources to exploit price fluctuation, and (iii) dynamic provisioning of generators - orchestrating which generators produce what portion of the energy demand. While prior literature has considered these techniques in isolation, we show how they can be used in coordination to manage both the supply and demand of power to achieve substantial cost reduction.

Notation	Definition
T	Number of time slots
N	Number of on-site generators
β_s	Switching cost of a server ($)
β_g	Startup cost of an on-site generator ($)
c_m	Sunk cost of maintaining a generator in its active state per slot ($)
c_o	Incremental cost for an active generator to output an additional unit of energy ($/Wh)
L	The maximum output of a generator (Watt)
$a(t)$	Workload at time t
$p(t)$	Price per unit energy drawn from the grid at t ($P_{\min} \le p(t) \le P_{\max}$) ($/Wh)
$x(t)$	Number of active servers at t
$s(t)$	Total server service capability at t
$v(t)$	Grid power used at t (Watt)
$y(t)$	Number of active on-site generators at t
$u(t)$	Total power output from active generators at t (Watt)
$g_t(x(t),a(t))$	Total power consumption as a function of $x(t)$ and $a(t)$ at t (Watt)

Note: we use bold symbols to denote vectors, *e.g.*, $\boldsymbol{x} = \langle x(t) \rangle$. Brackets indicate the unit.

Table 2: Key notation.

2.1 Model Assumptions

We adopt a discrete-time model whose time slot matches the timescale at which the scheduling decisions can be updated. Without loss of generality, we assume there are totally T slots, and each has a unit length.

Workload model. Similar to existing work [13, 34, 16], we consider a "mice" type of workload for the data center where each job has a small transaction size and short duration. Jobs arriving in a slot get served in the same slot. Workload can be split among active servers at arbitrary granularity like a fluid. These assumptions model a "request-response" type of workload that characterizes serving web

content or hosted application services that entail short but real-time interactions between the user and the server. The workload to be served at time t is represented by $a(t)$. Note that we do not rely on any specific stochastic model of $a(t)$.

Server model. We assume that the data center consists of a sufficient number of homogeneous servers, and each has unit service capacity, *i.e.*, it can serve at most one unit workload per slot, and the same power consumption model. Let $x(t)$ be the number of active servers and $s(t) \in [0, x(t)]$ be the total server service capability at time t. It is clear that $s(t)$ should be larger than $a(t)$ to get the workload served in the same slot. We model the aggregate server power consumption as $b(t) \triangleq f_s(x(t), s(t))$, an increasing and convex function of $x(t)$ and $s(t)$. That is, the first and second order partial derivatives in $x(t)$ and $s(t)$ are all non-negative. Since $f_s(x(t), s(t))$ is increasing in $s(t)$, it is optimal to always set $s(t) = a(t)$. Thus, we have $b(t) = f_s(x(t), a(t))$ and $x(t) \geq a(t)$.

This power consumption model is quite general and captures many common server models. One example is the commonly adopted standard linear model [9]:

$$f_s(x(t), a(t)) = c_{idle}x(t) + (c_{peak} - c_{idle})a(t),$$

where c_{idle} and c_{peak} are the power consumed by an server at idle and fully utilized state, respectively. Most servers today consume significant amounts of power even when idle. A holy grail for server design is to make them "power proportional" by making c_{idle} zero [32].

Besides, turning a server on entails switching cost [28], denoted as β_s, including the amortized service interruption cost, wear-and-tear cost, *e.g.*, component procurement, replacement cost (hard-disks in particular) and risk associated with server switching. It is comparable to the energy cost of running a server for several hours [23].

In addition to servers, power conditioning and cooling systems also consume a significant portion of power. The three[1] contribute about 94% of overall power consumption and their power draw vary drastically with server utilization [33]. Thus, it is important to model the power consumed by power conditioning and cooling systems.

Power conditioning system model. Power conditioning system usually includes power distribution units (PDUs) and uninterruptible power supplies (UPSs). PDUs transform the high voltage power distributed throughout the data center to voltage levels appropriate for servers. UPSs provides temporary power during outage. We model the power consumption of this system as $f_p(b(t))$, an increasing and convex function of the aggregate server power consumption $b(t)$.

This model is general and one example is a quadratic function adopted in a comprehensive study on the data center power consumption [33]: $f_p(b(t)) = C_1 + \pi_1 b^2(t)$, where $C_1 > 0$ and $\pi_1 > 0$ are constants depending on specific PDUs and UPSs.

Cooling system model. We model the power consumed by the cooling system as $f_c^t(b(t))$, a time-dependent (*e.g.*, depends on ambient weather conditions) increasing and convex function of $b(t)$.

This cooling model captures many common cooling systems. According to [24], the power consumption of an outside air cooling system can be modelled as a time-dependent cubic function of $b(t)$: $f_c^t(b(t)) = K_t b^3(t)$, where $K_t > 0$ depends on ambient weather conditions, such as air temperature, at time t. According to [33], the power draw of a water chiller cooling system can be modelled as a time-dependent quadratic function of $b(t)$: $f_c^t(b(t)) = Q_t b^2(t) + L_t b(t) + C_t$, where $Q_t, L_t, C_t \geq 0$ depend on outside air and chilled water temperature at time t. Note that all we need is $f_c^t(b(t))$ is increasing and convex in $b(t)$.

On-site generator model. We assume that the data center has N units of homogeneous on-site generators, each having an power output capacity L. Similar to generator models studied in the unit commitment problem [20], we define a generator startup cost β_g, which typically involves heating up cost, additional maintenance cost due to each startup (*e.g.*, fatigue and possible permanent damage resulted by stresses during startups), c_m as the sunk cost of maintaining a generator in its active state for a slot, and c_o as the incremental cost for an active generator to output an additional unit of energy. Thus, the total cost for $y(t)$ active generators that output $u(t)$ units of energy at time t is $c_m y(t) + c_o u(t)$.

Grid model. The grid supplies energy to the data center in an "on-demand" fashion, with time-varying price $p(t)$ per unit energy at time t. Thus, the cost of drawing $v(t)$ units of energy from the grid at time t is $p(t)v(t)$. Without loss of generality, we assume $0 \leq P_{\min} \leq p(t) \leq P_{\max}$.

To keep the study interesting and practically relevant, we make the following assumptions: (i) the server and generator turning-on cost are strictly positive, *i.e.*, $\beta_s > 0$ and $\beta_g > 0$. (ii) $c_o + c_m/L < P_{\max}$. This ensures that the minimum on-site energy price is cheaper than the maximum grid energy price. Otherwise, it should be clear that it is optimal to always buy energy from the grid, because in that case the grid energy is cheaper and incurs no startup costs.

2.2 Problem Formulation

Based on the above models, the data center total power consumption is the sum of the server, power conditioning system and the cooling system power draw, which can be expressed as a time-dependent function of $b(t)$ ($b(t) = f_s(x(t), a(t))$):

$$b(t) + f_p(b(t)) + f_c^t(b(t)) \triangleq g_t(x(t), a(t)).$$

We remark that $g_t(x(t), a(t))$ is increasing and convex in $x(t)$ and $a(t)$. This is because it is the sum of three increasing and convex functions. *Note that all results we derive in this paper apply to any $g_t(x, a)$ as long as it is increasing and convex in x and a.*

Our objective is to minimize the data center total cost in entire horizon $[1, T]$, which is given by

$$\text{Cost}(x, y, u, v) \triangleq \sum_{t=1}^{T} \{v(t)p(t) + c_o u(t) + c_m y(t) + \beta_s [x(t) - x(t-1)]^+ + \beta_g [y(t) - y(t-1)]^+\}, \quad (1)$$

which includes the cost of grid electricity, the running cost of on-site generators, and the switching cost of servers and on-site generators in the entire horizon $[1, T]$. Throughout this paper, we set initial condition $x(0) = y(0) = 0$.

We formally define the data center cost minimization problem as a non-linear mixed-integer program, given the workload $a(t)$, the grid price $p(t)$ and the time-dependent func-

[1]The other two, networking and lighting, consume little power and have less to do with server utilization. Thus, we do not model the two in this paper.

tion $g_t(x,a)$, for $1 \le t \le T$, as time-varying inputs.

$$\min_{x,y,u,v} \quad \text{Cost}(x,y,u,v) \tag{2}$$
$$\text{s.t.} \quad u(t)+v(t) \ge g_t(x(t),a(t)), \tag{3}$$
$$u(t) \le Ly(t), \tag{4}$$
$$x(t) \ge a(t), \tag{5}$$
$$y(t) \le N, \tag{6}$$
$$x(0)=y(0)=0, \tag{7}$$
$$\text{var} \quad x(t),y(t) \in \mathbb{N}^0, u(t),v(t) \in \mathbb{R}_0^+,\ t \in [1,T],$$

where $[\cdot]^+ = \max(0,\cdot)$, $\mathbb{N}^0$ and $\mathbb{R}_0^+$ represent the set of non-negative integers and real numbers, respectively.

Constraint (3) ensures the total power consumed by the data center is jointly supplied by the generators and the grid. Constraint (4) captures the maximal output of the on-site generator. Constraint (5) specifies that there are enough active servers to serve the workload. Constraint (6) is generator number constraint. Constraint (7) is the boundary condition.

Note that this problem is challenging to solve. First, it is a non-linear mixed-integer optimization problem. Further, the objective function values across different slots are correlated via the switching costs $\beta_s[x(t)-x(t-1)]^+$ and $\beta_g[y(t)-y(t-1)]^+$, and thus cannot be decomposed. Finally, to obtain an online solution we do not even know the inputs beyond current slot.

Next, we introduce a proposition to simplify the structure of the problem. Note that if $(x(t))_{t=1}^T$ and $(y(t))_{t=1}^T$ are given, the problem in (2)-(7) reduces to a linear program and can be solved independently for each slot. We then obtain the following.

PROPOSITION 1. *Given any $x(t)$ and $y(t)$, the $u(t)$ and $v(t)$ that minimize the cost in (2) with any $g_t(x,a)$ that is increasing in x and a, are given by: $\forall t \in [1,T]$,*

$$u(t) = \begin{cases} 0, & \text{if } p(t) \le c_o, \\ \min\left(Ly(t), g_t(x(t),a(t))\right), & \text{otherwise,} \end{cases}$$

and

$$v(t) = g_t(x(t),a(t)) - u(t).$$

Note that $u(t), v(t)$ can be computed using *only* $x(t), y(t)$ at current time t, thus can be determined in an online fashion.

Intuitively, the above proposition says if the on-site energy price c_o is higher than the grid price $p(t)$, we should buy energy from the grid; otherwise, it is the best to buy the cheap on-site energy up to its maximum supply $L \cdot y(t)$ and the rest (if any) from the more expensive grid. With the above proposition, we can reduce the non-linear mixed-integer program in (2)-(7) with variables $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ to the following integer program with only variables $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\textbf{DCM}:$$
$$\min \quad \sum_{t=1}^{T} \big\{\psi\left(y(t),p(t),d_t(x(t))\right) + \beta_s[x(t)-x(t-1)]^+ + \beta_g[y(t)-y(t-1)]^+\big\} \tag{8}$$
$$\text{s.t.} \quad x(t) \ge a(t),$$
$$(6),(7),$$
$$\text{var} \quad x(t),y(t) \in \mathbb{N}^0,\ t \in [1,T],$$

where $d_t(x(t)) \triangleq g_t(x(t),a(t))$, for the ease of presentation in later sections, is increasing and convex in $x(t)$ and $\psi\left(y(t),p(t),d_t(x(t))\right)$ replaces the term $v(t)p(t)+c_o u(t)+c_m y(t)$ in the original cost function in (2) and is defined as

$$\psi\left(y(t),p(t),d_t(x(t))\right) \tag{9}$$
$$\triangleq \begin{cases} c_m y(t) + p(t) d_t(x(t)), & \text{if } p(t) \le c_o, \\ c_m y(t) + c_o L y(t) + & \text{if } p(t) > c_o \text{ and} \\ p(t)\left(d_t(x(t)) - Ly(t)\right), & d_t(x(t)) > Ly(t), \\ c_m y(t) + c_o d_t(x(t)), & \text{else.} \end{cases}$$

As a result of the analysis above, it suffices to solve the above formulation of **DCM** with only variables $\boldsymbol{x}$ and $\boldsymbol{y}$, in order to minimize the data center operating cost.

2.3 An Offline Optimal Algorithm

We present an offline optimal algorithm for solving problem **DCM** using Dijkstra's shortest path algorithm [15]. We construct a graph $G=(V,E)$, where each vertex denoted by the tuple $\langle x,y,t\rangle$ represents a state of the data center where there are x active servers, and y active generators at time t. We draw a directed edge from each vertex $\langle x(t-1),y(t-1),t-1\rangle$ to each possible vertex $\langle x(t),y(t),t\rangle$ to represent the fact that the data center can transit from the first state to the second state. Further, we associate the cost of that transition shown below as the weight of the edge:

$$\psi\left(y(t),p(t),d_t(x(t))\right) + \beta_s[x(t)-x(t-1)]^+ + \beta_g[y(t)-y(t-1)]^+.$$

Next, we find the minimum weighted path from the initial state represented by vertex $\langle 0,0,0\rangle$ to the final state represented by vertex $\langle 0,0,T+1\rangle$ by running Dijkstra's algorithm on graph G. Since the weights represent the transition costs, it is clear that finding the minimum weighted path in G is equivalent to minimizing the total transitional costs. Thus, our offline algorithm provides an optimal solution for problem **DCM**.

THEOREM 1. *The algorithm described above finds an optimal solution to problem **DCM** in time $O\left(M^2N^2T\log\left(MNT\right)\right)$, where T is the number of slots, N the number of generators and $M = \max_{1\le t\le T}\lceil a(t)\rceil$.*

PROOF. Since the numbers of active servers and generators are at most M and N, respectively, and there are $T+2$ time slots, graph G has $O(MNT)$ vertices and $O(M^2N^2T)$ edges. Thus, the run time of Dijkstra's algorithm on graph G is $O\left(M^2N^2T\log\left(MNT\right)\right)$. □

Remark: In practice, the time-varying input sequences ($p(t)$, $a(t)$ and g_t) may not be available in advance and hence it may be difficult to apply the above offline algorithm. However, an offline optimal algorithm can serve as a benchmark, using which we can evaluate the performance of online algorithms.

3. THE BENEFIT OF JOINT OPTIMIZATION

Data center cost minimization (**DCM**) entails the joint optimization of both server capacity that determines the energy demand and on-site power generation that determines the energy supply. Now consider the situation where the data center optimizes the energy demand and supply separately.

First, the data center dynamically provisions the server capacity according to the grid power price $p(t)$. More formally, it solves the *capacity provisioning* problem which we refer to as **CP** below.

$$\begin{aligned} \mathbf{CP}: \quad \min \quad & \sum_{t=1}^{T} \left\{ p(t) \cdot d_t(x(t)) + \beta_s [x(t) - x(t-1)]^+ \right\} \\ \text{s.t.} \quad & x(t) \geq a(t), \\ & x(0) = 0, \\ \text{var} \quad & x(t) \in \mathbb{N}^0,\ t \in [1, T]. \end{aligned}$$

Solving problem **CP** yields $\bar{\boldsymbol{x}}$. Thus, the total power demand at time t given $\bar{x}(t)$ is $d_t(\bar{x}(t))$. Note that $d_t(\bar{x}(t))$ is not just server power consumption, but also includes consumption of power conditioning and cooling systems, as described in Sec. 2.2.

Second, the data center minimizes the cost of satisfying the power demand due to $d_t(\bar{x}(t))$, using both the grid and the on-site generators. Specifically, it solves the *energy provisioning* problem which we refer to as **EP** below.

$$\begin{aligned} & \mathbf{EP}: \\ \min \quad & \sum_{t=1}^{T} \left\{ \psi\left(y(t), p(t), d_t(\bar{x}(t))\right) + \beta_g [y(t) - y(t-1)]^+ \right\} \\ & y(0) = 0, \\ \text{var} \quad & y(t) \in \mathbb{N}^0,\ t \in [1, T]. \end{aligned}$$

Let $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ be the solution obtained by solving **CP** and **EP** separately in sequence and $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be the solution obtained by solving the joint-optimization **DCM**. Further, let $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}, \boldsymbol{y})$ be the value of the data center's total cost for solution $(\boldsymbol{x}, \boldsymbol{y})$, including both generator and server costs as represented by the objective function (8) of problem **DCM**. The additional benefit of joint optimization over optimizing independently is simply the relationship between $\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ and $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*)$. It is clear that $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ obeys all the constraints of **DCM** and hence is a feasible solution of **DCM**. Thus, $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq \mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$. We can measure the factor loss in optimality ρ due to optimizing separately as opposed to optimizing jointly on the worst-case input as follows:

$$\rho \triangleq \max_{\text{all inputs}} \frac{\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})}{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*)}.$$

The following theorem characterizes the benefit of joint optimization over optimizing independently.

THEOREM 2. *The factor loss in optimality ρ by solving the problem **CP** and **EP** in sequence as opposed to optimizing jointly is given by $\rho = LP_{\max}/(Lc_o + c_m)$ and it is tight.*

The above theorem guarantees that for *any* time duration T, *any* workload $\boldsymbol{a}$, *any* grid price $\boldsymbol{p}$ and *any* function $g_t(x, a)$ as long as it is increasing and convex in x and a, solving problem **DCM** by first solving **CP** then solving **EP** in sequence yields a solution that is within a factor $LP_{\max}/(Lc_o + c_m)$ of solving **DCM** directly. Further, the ratio is tight in that there exists an input to **DCM** where the ratio $\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})/\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*)$ equals $LP_{\max}/(Lc_o + c_m)$.

The theorem shows in a quantitative way that a larger price discrepancy between the maximum grid price and the on-site power yields a larger gain by optimizing the energy provisioning and capacity provisioning jointly. Over the past decade, utilities have been exposing a greater level of grid price variation to their customers with mechanisms such as time-of-use pricing where grid prices are much more expensive during peak hours than during the off-peak periods. This likely leads to larger price discrepancy between the grid and the on-site power. In that case, our result implies that a joint optimization of power and server resources is likely to yield more benefits to a hybrid data center.

Besides characterizing the benefit of jointly optimizing power and server resources, the decomposition of problem **DCM** into problems **CP** and **EP** provides a key approach for our online algorithm design. Problem **DCM** has an objective function with mutually-dependent coupled variables $\boldsymbol{x}$ and $\boldsymbol{y}$ indicating the server and generator states, respectively. This coupling (specifically through the function $\psi\left(y(t), p(t), d_t(x(t))\right)$) makes it difficult to design provably good online algorithms. However, instead of solving problem **DCM** directly, we devise online algorithms to solve problems **CP** that involves only server variable $\boldsymbol{x}$ and **EP** that involves only the generator variables $\boldsymbol{y}$. Combining the online algorithms for **CP** and **EP** respectively yields the desired online algorithm for **DCM**.

	Cooling & Power Conditioning	**Optimization Type**	**Competitive Ratio**
LCP [23]	No	obj: convex var: continuous	3
CSR [27]	No	obj: linear var: integer	$2 - \alpha_s$
GCSR this work	Yes	obj: convex and increasing var: integer	$2 - \alpha_s$

Note that α_s is the normalized look-ahead window size, whose representations are different under the different settings of [27] and our work.

Table 3: Comparison of the algorithm GCSR proposed in this paper, CSR in [27], and LCP in [23].

4. ONLINE ALGORITHMS FOR ON-GRID DATA CENTERS

We first develop an online algorithm for **DCM** for an *on-grid* data center, where there is no on-site power generation, a scenario that captures most data centers today. Since *on-grid* data center has no on-site power generation, solving **DCM** for it reduces to solving problem **CP** described in Sec. 3.

Problems of this kind have been studied in the literature (see *e.g.*, [23, 27]). The difference of our work from [23, 27] is as follows (also summarized in Table 3). From the modelling aspect, we explicitly take into account power consumption of both cooling and power conditioning systems, in addition to servers. From the formulation aspect, we are solving a different optimization problem, *i.e.*, an integer program with convex and increasing objective function. From the theoretical result aspects, we achieve a small competitive ratio of $2 - \alpha_s$, which quickly decreases to 1 as look-ahead window w increase.

Recall that **CP** takes as input the workload $\boldsymbol{a}$, the grid price $\boldsymbol{p}$ and the time-dependent function g_t, $\forall t$ and outputs the number of active servers $\boldsymbol{x}$. We construct solutions to **CP** in a divide-and-conquer fashion. We will first de-

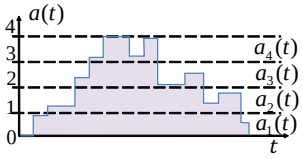


Figure 2: An example of how workload $\boldsymbol{a}$ is decomposed into 4 sub-demands.

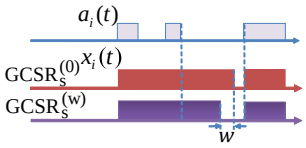


Figure 3: An example of $a_i(t)$ and corresponding solution obtained by $\mathbf{GCSR_s^{(w)}}$.

compose the demand $\boldsymbol{a}$ into sub-demands and define corresponding sub-problem for each server, and then solve capacity provisioning *separately* for each sub-problem. Note that the key is to correctly decompose the demand and define the subproblems so that the combined solution is still optimal. More specifically, we slice the demand as follows: for $1 \le i \le M = \max_{1\le t\le T} \lceil a(t) \rceil$, $1 \le t \le T$,

$$a_i(t) \triangleq \min\left\{1, \max\left\{0, a(t) - (i-1)\right\}\right\}.$$

And the corresponding sub-problem $\mathbf{CP_i}$ is defined as follows.

$$\begin{aligned} \mathbf{CP_i}: \quad \min \quad & \sum_{t=1}^{T} \left\{ p(t) \cdot d_t^i \cdot x_i(t) + \beta_s [x_i(t) - x_i(t-1)]^+ \right\} \\ \text{s.t.} \quad & x_i(t) \ge a_i(t), \\ & x_i(0) = 0, \\ \text{var} \quad & x_i(t) \in \{0,1\},\ t \in [1,T], \end{aligned}$$

where $x_i(t)$ indicates whether the i-th server is on at time t and $d_t^i \triangleq d_t(i) - d_t(i-1)$. d_t^i can be interpreted as the power consumption due to the i-th server at t.

Problem $\mathbf{CP_i}$ solves the capacity provisioning problem with inputs workload $\boldsymbol{a_i}$, grid price $\boldsymbol{p}$ and d_t^i. The key reason for our decomposition is that $\mathbf{CP_i}$ is easier to solve, since $\boldsymbol{a_i}$ take values in $[0,1]$ and exactly one server is required to serve each $\boldsymbol{a_i}$. Generally speaking, a divide-and-conquer manner may suffer from optimality loss. Surprisingly, as the following theorem states, the individual optimal solutions for problems $\mathbf{CP_i}$ can be put together to form an optimal solution to the original problem $\mathbf{CP}$. Denote $\text{C}_{\text{CP}_\text{i}}(\boldsymbol{x_i})$ as the cost of solution $\boldsymbol{x_i}$ for problem $\mathbf{CP_i}$ and $\text{C}_{\text{CP}}(\boldsymbol{x})$ the cost of solution $\boldsymbol{x}$ for problem $\mathbf{CP}$.

THEOREM 3. *Consider problem* $\mathbf{CP}$ *with any* $d_t(x(t)) = g_t(x(t), a(t)))$ *that is convex in* $x(t)$. *Let* $\bar{\boldsymbol{x}}_i$ *be an optimal solution and* $\boldsymbol{x}_i^{on}$ *an online solution for problem* $\mathbf{CP_i}$ *with workload* $\boldsymbol{a}_i$, *then* $\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i$ *is an optimal solution for* $\mathbf{CP}$ *with workload* $\boldsymbol{a}$. *Furthermore, if* $\forall \boldsymbol{a}_i, i$, *we have* $\text{C}_{\text{CP}_\text{i}}(\boldsymbol{x}_i^{on}) \le \gamma \cdot \text{C}_{\text{CP}_\text{i}}(\bar{\boldsymbol{x}}_i)$ *for a constant* $\gamma \ge 1$, *then* $\text{C}_{\text{CP}}(\sum_{i=1}^{M} \boldsymbol{x}_i^{on}) \le \gamma \cdot \text{C}_{\text{CP}}(\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i)$, $\forall \boldsymbol{a}$.

Thus, it remains to design algorithms for each $\mathbf{CP_i}$. To solve $\mathbf{CP_i}$ in an online fashion one need only orchestrate one server to satisfy the workload $\boldsymbol{a}_i$ and minimize the total cost. When $a_i(t) > 0$, we must keep the server active to satisfy the workload. The challenging part is what we should do if the server is already active but $a_i(t) = 0$. Should we turn off the server immediately or keep it idling for some time? Should we distinguish the scenarios when the grid price is high versus low?

Inspired by "ski-rental" [12] and [27], we solve $\mathbf{CP_i}$ by the following "break-even" idea. During the idle period, *i.e.*, $a_i(t) = 0$, we accumulate an "idling cost" and when it reaches β_s, we turn off the server; otherwise, we keep the server idling. Specifically, our online algorithm $\mathbf{GCSR_s^{(w)}}$ (Generalized Collective Server Rental) for $\mathbf{CP_i}$ has a look-ahead window w. At time t, if there exist $\tau' \in [t, t+w]$ such that the idling cost till τ' is at least β_s, we turn off the server; otherwise, we keep it idling. More formally, we have Algorithm 1 and its competitive analysis in Theorem 4. A simple example of $\mathbf{GCSR_s^{(w)}}$ is shown in Fig. 3.

Algorithm 1 $\mathbf{GCSR_s^{(w)}}$ for problem $\mathbf{CP_i}$

$C_i = 0, x_i(0) = 0$
at current time t, **do**
  Set $\tau' \leftarrow \min\{t' \in [t, t+w] \mid C_i + \sum_{\tau=t}^{t'} p(\tau) d_\tau^i \ge \beta_s\}$
  if $a_i(t) > 0$ **then**
    $x_i(t) = 1$ and $C_i = 0$
  else if $\tau' =$ NULL or $\exists \tau \in [t, \tau']$, $a_i(\tau) > 0$ **then**
    $x_i(t) = x_i(t-1)$ and $C_i = C_i + p(t) d_t^i x_i(t)$
  else
    $x_i(t) = 0$ and $C_i = 0$
  end if

Our online algorithm for $\mathbf{CP}$, denoted as $\mathbf{GCSR^{(w)}}$, first employs $\mathbf{GCSR_s^{(w)}}$ to solve each $\mathbf{CP_i}$ on workload $\boldsymbol{a}_i$, $1 \le i \le M$, in an online fashion to produce output $\boldsymbol{x}_i^{on}$ and then simply outputs $\sum_{i=1}^{M} \boldsymbol{x}_i^{on} = \boldsymbol{x}^{on}$ as the output for the original problem $\mathbf{CP}$.

THEOREM 4. $\mathbf{GCSR_s^{(w)}}$ *achieves a competitive ratio of* $2 - \alpha_s$ *for* $\mathbf{CP_i}$, *where* $\alpha_s \triangleq \min(1, w d_{\min} P_{\min}/\beta_s) \in [0,1]$ *is a "normalized" look-ahead window size and* $d_{\min} \triangleq \min_t\{d_t(1) - d_t(0)\}$. *Hence, according to Theorem 3,* $\mathbf{GCSR^{(w)}}$ *achieves the same competitive ratio for* $\mathbf{CP}$. *Further, no deterministic online algorithm with a look-ahead window* w *can achieve a smaller competitive ratio.*

A consequence of Theorem 4 is that when the look-ahead window size w reaches a break-even interval $\Delta_s \triangleq \beta_s/(d_{\min} P_{\min})$, our online algorithm has a competitive ratio of 1. That is, having a look-ahead window larger than Δ_s will not decrease the cost any further.

5. ONLINE ALGORITHMS FOR HYBRID DATA CENTERS

Unlike on-grid data centers, hybrid data centers have on-site power generation and therefore have to solve both capacity provisioning ($\mathbf{CP}$) and energy provisioning ($\mathbf{EP}$) to solve the data center cost minimization ($\mathbf{DCM}$) problem. We design an online algorithm that we call $\mathbf{DCMON}$ solving $\mathbf{DCM}$ as follows.

1. Run algorithm $\mathbf{GCSR}$ from Sec. 4 to solve $\mathbf{CP}$ that takes workload $\boldsymbol{a}$, grid price $\boldsymbol{p}$ and time-dependent function g_t, $\forall t$ as input and produces the number of active servers $\boldsymbol{x}^{on}$.

2. Run algorithm $\mathbf{CHASE}$ described in Section 5.2 below to solve $\mathbf{EP}$ that takes the energy demand $d_t(x^{on}(t)) = g_t(x^{on}(t), a(t))$ and grid price $p(t)$, $\forall t$ as input and decides when to turn on/off on-site generators and how much power to draw from the generators and the grid. Note that a similar problem has been studied in the

microgrid scenarios for energy generation scheduling in our previous work [26]. In this paper, we adapt algorithm **CHASE** developed in [26] to our data center scenarios to solve **EP** in an online fashion.

For the sake of completeness, we first briefly present the design behind **CHASE** in Sec. 5.1 and the algorithm and its intuitions in Sec. 5.2. Then we present the combined algorithm **DCMON** in Sec. 5.3.

5.1 A useful structure of an offline optimal solution of EP

We first reveal an elegant structure of an offline optimal solution and then exploit this structure in the design of our online algorithm **CHASE**.

5.1.1 *Decompose EP into sub-problems EP_is*

For the ease of presentation, we denote $e(t) = d_t(x^{on}(t))$. Similar as the decomposition of workload when solving **CP**, we decompose the energy demand $\boldsymbol{e}$ into N sub-demands and define sub-problem for each generator, then solve energy provisioning *separately* for each sub-problem, where N is the number of on-site generators. Specifically, for $1 \le i \le N$, $1 \le t \le T$,

$$e_i(t) \triangleq \min\{L, \max\{0, e(t) - (i-1)L\}\}.$$

The corresponding sub-problem $\mathbf{EP_i}$ is in the same form as **EP** except that $d_t(\bar{x}(t))$ is replaced by $e_i(t)$ and $y(t)$ is replaced by $y_i(t) \in \{0,1\}$. Using this decomposition, we can solve **EP** on input $\boldsymbol{e}$ by simultaneously solving simpler problems $\mathbf{EP_i}$ on input $\boldsymbol{e}_i$ that only involve a single generator. Theorem 5 shows that the decomposition incurs no optimality loss. Denote $\mathrm{C}_{\mathrm{EP_i}}(\boldsymbol{y_i})$ as the cost of solution $\boldsymbol{y_i}$ for problem $\mathbf{EP_i}$ and $\mathrm{C}_{\mathrm{EP}}(\boldsymbol{y})$ the cost of solution $\boldsymbol{y}$ for problem **EP**.

THEOREM 5. *Let $\bar{\boldsymbol{y}}_i$ be an optimal solution and $\boldsymbol{y}_i^{on}$ an online solution for $\mathbf{EP_i}$ with energy demand $\boldsymbol{e}_i$, then $\sum_{i=1}^N \bar{\boldsymbol{y}}_i$ is an optimal solution for* **EP** *with energy demand $\boldsymbol{e}$. Furthermore, if $\forall \boldsymbol{e}_i, i$, we have $\mathrm{C}_{\mathrm{EP_i}}(\boldsymbol{y}_i^{on}) \le \gamma \cdot \mathrm{C}_{\mathrm{EP_i}}(\bar{\boldsymbol{y}}_i)$ for a constant $\gamma \ge 1$, then $\mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^N \boldsymbol{y}_i^{on}) \le \gamma \cdot \mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^N \bar{\boldsymbol{y}}_i), \forall \boldsymbol{e}$.*

5.1.2 *Solve each sub-problem EP_i*

Based on Theorem 5, it remains to design algorithms for each $\mathbf{EP_i}$. Define

$$r_i(t) = \psi(0, p(t), e_i(t)) - \psi(1, p(t), e_i(t)). \quad (10)$$

$r_i(t)$ can be interpreted as the one-slot cost difference between not using and using on-site generation. Intuitively, if $r_i(t) > 0$ (resp. $r_i(t) < 0$), it will be desirable to turn on (resp. off) the generator. However, due to the startup cost, we should not turn on and off the generator too frequently. Instead, we should evaluate whether the *cumulative* gain or loss in the future can offset the startup cost. This intuition motivates us to define the following cumulative cost difference $R_i(t)$. We set initial values as $R_i(0) = -\beta_g$ and define $R_i(t)$ inductively:

$$R_i(t) \triangleq \min\{0, \max\{-\beta_g, R_i(t-1) + r_i(t)\}\}, \quad (11)$$

Note that $R_i(t)$ is only within the range $[-\beta_g, 0]$. An important feature of $R_i(t)$ useful later in online algorithm design is that it can be computed given the past and current inputs. An illustrating example of $R_i(t)$ is shown in Fig. 4.

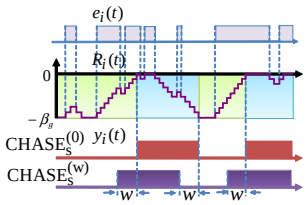


Figure 4: An example of $e_i(t)$, $R_i(t)$ and the corresponding solution obtained by $\mathbf{CHASE_s^{(w)}}$ for $\mathbf{EP_i}$.

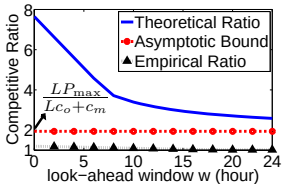


Figure 5: Theoretical and empirical ratios of algorithm $\mathbf{DCMON^{(w)}}$ vs. look-ahead window size w.

Intuitively, when $R_i(t)$ hits its boundary 0, the cost difference between not using and using on-site generation within a certain period is at least β_g, which can offset the startup cost. Thus, it makes sense to turn on the generator. Similarly, when $R_i(t)$ hits $-\beta_g$, it may be better to turn off the generator and use the grid. The following theorem formalizes this intuition, and shows an optimal solution $\bar{y}_i(t)$ for problem $\mathbf{EP_i}$ at the time epoch when $R_i(t)$ hits its boundary values $-\beta_g$ or 0.

THEOREM 6. *There exists an offline optimal solution for problem $\mathbf{EP_i}$, denoted by $\bar{y}_i(t)$, $1 \le t \le T$, so that:*

- *if $R_i(t) = -\beta_g$, then $\bar{y}_i(t) = 0$;*
- *if $R_i(t) = 0$, then $\bar{y}_i(t) = 1$.*

5.2 Online algorithm CHASE

Our online algorithm $\mathbf{CHASE_s^{(w)}}$ with look-ahead window w exploits the insights revealed in Theorem 6 to solve $\mathbf{EP_i}$. The idea behind $\mathbf{CHASE_s^{(w)}}$ is to track the offline optimal in an online fashion. In particular, at time 0, $R_i(0) = -\beta_g$ and we set $y_i(t) = 0$. We keep tracking the value of $R_i(t)$ at every time slot within the look-ahead window. Once we observe that $R_i(t)$ hits values $-\beta_g$ or 0, we set the $y_i(t)$ to the optimal solution as Theorem 6 reveals; otherwise, keep $y_i(t) = y_i(t-1)$ unchanged. More formally, we have Algorithm 2 and its competitive analysis in Theorem 7. An example of $\mathbf{CHASE_s^{(w)}}$ is shown in Fig. 4.

The online algorithm for **EP**, denoted as $\mathbf{CHASE^{(w)}}$, first employs $\mathbf{CHASE_s^{(w)}}$ to solve each $\mathbf{EP_i}$ on energy demand $\boldsymbol{e}_i$, $1 \le i \le N$, in an online fashion to produce output $\boldsymbol{y}_i^{on}$ and then simply outputs $\sum_{i=1}^N \boldsymbol{y}_i^{on}$ as the output for the original problem **EP**.

Algorithm 2 $\mathbf{CHASE_s^{(w)}}$ for problem $\mathbf{EP_i}$

```
at current time t, do
Obtain (R_i(τ))_{τ=t}^{t+w}
Set τ' ← min{τ ∈ [t, t+w] | R_i(τ) = 0 or −β_g}
if τ' = NULL then
    y_i(t) = y_i(t−1)
else if R_i(τ') = 0 then
    y_i(t) = 1
else
    y_i(t) = 0
end if
```

THEOREM 7. $\mathbf{CHASE_s^{(w)}}$ *for problem $\mathbf{EP_i}$ with a look-*

ahead window w has a competitive ratio of

$$1+\frac{2\beta_g\left(LP_{\max}-Lc_o-c_m\right)}{\beta_g LP_{\max}+wc_mP_{\max}\left(L-\frac{c_m}{P_{\max}-c_o}\right)}.$$

Hence, according to Theorem 5, $\mathbf{CHASE}^{(\mathbf{w})}$ *achieves the same competitive ratio for problem* **EP**.

5.3 Combining GCSR and CHASE

Our algorithm $\mathbf{DCMON}^{(\mathbf{w})}$ for solving problem **DCM** with a look-ahead window of $w \geq 0$, *i.e.*, knowing grid prices $p(\tau)$, workload $a(\tau)$ and the function $g_\tau, 1 \leq \tau \leq t+w$, at time t, first uses **GCSR** from Sec. 4 to solve problem **CP** and then uses **CHASE** in Sec. 5.2 to solve problem **EP**. An important observation is that the available look-ahead window size for **GCSR** to solve **CP** is w, *i.e.*, knows $p(\tau)$, $a(\tau)$ and g_τ, $1 \leq \tau \leq t+w$, at time t; however, the available look-ahead window size for **CHASE** to solve **EP** is only $[w-\Delta_s]^+$, *i.e.*, knows $p(\tau)$ and $e(\tau)=d_\tau(x^{on}(\tau))$, $1 \leq \tau \leq t+[w-\Delta_s]^+$, at time t (Δ_s is the break-even interval defined in Sec. 4). Detailed explanation on this is relegated to our technical report [39].

Thus, a bound on the competitive ratio of $\mathbf{DCMON}^{(\mathbf{w})}$ is the product of competitive ratios for $\mathbf{GCSR}^{(\mathbf{w})}$ and $\mathbf{CHASE}^{([\mathbf{w}-\mathbf{\Delta_s}]^+)}$ from Theorems 4 and 7, respectively, and the optimality loss ratio $LP_{\max}/\left(Lc_o+c_m\right)$ due to the offline-decomposition stated in Sec. 3, which is given in the following Theorem.

THEOREM 8. $\mathbf{DCMON}^{(\mathbf{w})}$ *for problem* ***DCM*** *has a competitive ratio of*

$$\frac{P_{\max}\left(2-\alpha_s\right)}{c_o+c_m/L}\left[1+\frac{2\left(LP_{\max}-Lc_o-c_m\right)}{LP_{\max}+\alpha_gP_{\max}\left(L-\frac{c_m}{P_{\max}-c_o}\right)}\right]. \tag{12}$$

The ratio is also upper-bounded by

$$\frac{P_{\max}\left(2-\alpha_s\right)}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}}\cdot\frac{1}{1+\alpha_g}\right],$$

where $\alpha_s=\min\left(1,w/\Delta_s\right)\in[0,1]$ *and* $\alpha_g \triangleq \frac{c_m}{\beta_g}\left[w-\Delta_s\right]^+ \in [0,+\infty)$ *are "normalized" look-ahead window sizes.*

As the look-ahead window size w increases, the competitive ratio in Theorem 8 decreases to $LP_{\max}/\left(Lc_o+c_m\right)$ (c.f. Fig. 5), the inherent approximation ratio introduced by our offline decomposition approach discussed in Section 3. However, the real trace based empirical performance of $\mathbf{DCMON}^{(\mathbf{w})}$ without look-ahead is already close to the offline optimal, *i.e.*, ratio close to 1 (c.f. Fig. 5).

6. EMPIRICAL EVALUATION

We evaluate the performance of our algorithms by simulations based on real-world traces with the aim of (i) corroborating the empirical performance of our online algorithms under various realistic settings and the impact of having look-ahead information, (ii) understanding the benefit of opportunistically procuring energy from both on-site generators and the grid, as compared to the current practice of purchasing from the grid alone, (iii) studying how much on-site energy is needed for substantial cost benefits.

6.1 Parameters and Settings

Workload trace: We use the workload traces from the Akamai network [1, 30] that is the currently the world's largest content delivery network. The traces measure the workload of Akamai servers serving web content to actual end-users. Note that our workload is of the "request-and-response" type that we model in our paper. We use traces from the Akamai servers deployed in the New York and San Jose data centers that record the hourly average load served by each deployed server over 22 days from Dec. 21, 2008 to Jan. 11, 2009. The New York trace represents 2.5K servers that served about 1.4×10^{10} requests and 1.7×10^{13} bytes of content to end-users during our measurement period. The San Jose trace represents 1.5K servers that served about 5.5×10^9 requests and 8×10^{12} bytes of content. We show the workload in Fig. 6, in which we normalize the load by the server's service capacity. The workload is quite characteristic in that it shows daily variations (peak versus off-peak) and weekly variations (weekday versus weekend).

Grid price: We use traces of hourly grid power prices in New York [2] and San Jose [3] for the same time period, so that it can be matched up with the workload traces (c.f. Fig. 6). Both workload and grid price traces show strong diurnal properties: in the daytime, the workload and the grid price are relatively high; at night, on the contrary, both are low. This indicates the feasibility of reducing the data center cost by using the energy from the on-site generators during the daytime and use the grid at night.

Server model: As mentioned in Sec. 2, we assume the data center has a sufficient number of homogeneous servers to serve the incoming workload at any given time. Similar to a typical setting in [32], we use the standard linear server power consumption model. We assume that each server consumes 0.25KWh power per hour at full capacity and has a power proportional factor (PPF=$(c_{peak}-c_{idle})/c_{peak}$) of 0.6, which gives us $c_{idle}=0.1KW$, $c_{peak}=0.25KW$. In addition, we assume the server switching cost equals the energy cost of running a server for 3 hours. If we assume an average grid price as the price of energy, we get about $\beta_s=\$0.08$.

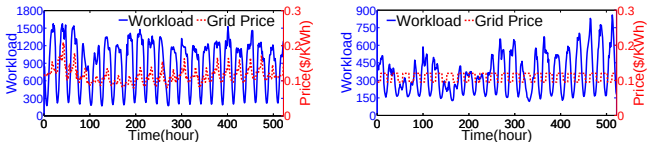
Cooling and power conditioning system model: We consider a water chiller cooling system. According to [5], during this 22-day winter period the average high and low temperatures of New York are $41^\circ F$ and $29^\circ F$, respectively. Those of San Jose are $58^\circ F$ and $41^\circ F$, respectively. Without loss of generality, we take the high temperature as the daytime temperature and the low temperature as the nighttime temperature. Thus, according to [33], the power consumed by water chiller cooling systems of the New York and San Jose data centers are about

$$f^t_{c,NY}(b)=\begin{cases}(0.041b^2+0.144b+0.047)b_{\max}, & \text{at daytime,}\\ (0.03b^2+0.136b+0.042)b_{\max}, & \text{at nighttime,}\end{cases}$$

and

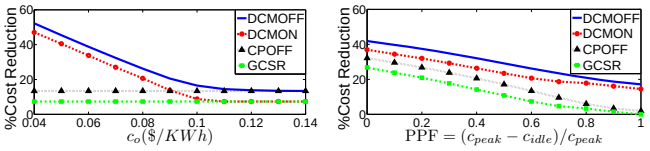
$$f^t_{c,SJ}(b)=\begin{cases}(0.06b^2+0.16b+0.054)b_{\max}, & \text{at daytime,}\\ (0.041b^2+0.144b+0.047)b_{\max}, & \text{at nighttime,}\end{cases}$$

where $b_{\max}$ is the maximum server power consumption and b is the server power consumption normalized by $b_{\max}$. The maximum server power consumption of the New York and San Jose data centers are $b^{NY}_{\max}=2500\times0.25=625KW$ and $b^{SJ}_{\max}=1500\times0.25=375KW$. Besides, the power con-



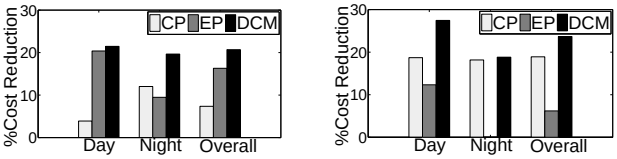
(a) New York (b) San Jose

Figure 6: Real-world workload from Akamai and the grid power price.



(a) Cost Reduction vs. c_o (b) Cost Reduction vs. PPF

Figure 7: Variation of cost reduction with model parameters.



(a) New York (b) San Jose

Figure 8: Relative values of CP, EP, and DCM.

sumed by the power conditioning system, including PDUs and UPSs, is $f_p(b) = (0.012b^2 + 0.046b + 0.056)b_{\max}$ [33].

Generator model: We adopt generators with specifications the same as the one in [4]. The maximum output of the generator is 60KW, *i.e.*, $L = 60KW$. The incremental cost to generate an additional unit of energy c_o is set to be \$0.08/KWh, which is calculated according to the gas price [2] and the generator efficiency [4]. Similar to [37], we set the sunk cost of running the generator for unit time c_m = \$1.2 and the startup cost β_g equivalent to the amortized capital cost, which gives β_g = \$24. Besides, we assume the number of generators $N = 10$, which is enough to satisfy all the energy demand for this trace and model we use.

Cost benchmark: Current data centers usually do not use dynamic capacity provisioning and on-site generators. Thus, we use the cost incurred by static capacity provisioning with grid power as the benchmark using which we evaluate the cost reduction due to our algorithms. Static capacity provisioning runs a fixed number of servers at all times to serve the workload, without dynamically turning on/off the servers. For our benchmark, we assume that the data center has complete workload information ahead of time and provisions exactly to satisfy the peak workload and uses only grid power. Using such a benchmark gives us a conservative evaluation of the cost saving from our algorithms.

Comparisons of Algorithms: We compare four algorithms: our online and offline optimal algorithms in on-grid scenarios, *i.e.*, **GCSR** and **CPOFF**, and hybrid scenarios, *i.e.*, **DCMON** and **DCMOFF**.

6.2 Impact of Model Parameters on Cost Reduction

We study the cost reduction provided by our offline and online algorithms for both on-grid and hybrid data centers using the New York trace unless specified otherwise. We assume no look-ahead information is available when running the online algorithms. We compute the cost reduction (in percentage) as compared to the cost benchmark which we described earlier. When all parameters take their default values, our offline (resp. online) algorithms provide up to 12.3% (resp., 7.3%) cost reduction for on-grid and 25.8% (resp., 20.7%) cost reduction for hybrid data centers (c.f. Fig. 7. The default value of c_o is \$0.08/KWh.). Note that the online algorithms provide cost reduction that are 5% smaller than offline algorithms on account of their lack of knowledge of future inputs. Further, note that cost reduction of a hybrid data center is larger than that of a on-grid data center, since hybrid data center has the ability to generate energy on-site to avoid higher grid prices. Nevertheless, the extent of cost reduction in all cases is high providing strong evidence for the need to perform energy and server capacity optimizations.

Data centers may deploy different types of servers and generators with different model parameters. It is then important to understand the impact on cost reduction due to these parameters. We first study the impact of varying c_o (c.f. Fig. 7). For a hybrid data center, as c_o increases the cost of on-site generation increases making it less effective for cost reduction (c.f Fig. 7a). For the same reason, the cost reduction of a hybrid data center tends to that of the on-grid data center with increasing c_o as on-site generation becomes less economical.

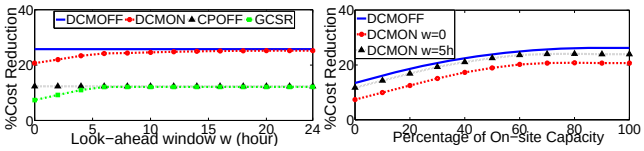
We then study the impact of power proportional factor (PPF). More specifically, we fix $c_{peak} = 0.25KW$, and vary PPF from 0 to 1 (c.f. Fig. 7b). As PPF increases, the server idle power decreases, thus dynamic provisioning has lesser impact on the cost reduction. This explains why **CP** achieves no cost reduction when PPF=1. Since **DCM** also solves **CP** problem, its performance degrades with increasing PPF as well.

6.3 The Relative Value of Energy versus Capacity Provisioning

In this subsection, we use both New York and San Jose traces. For a hybrid data center, we ask which optimization provides a larger cost reduction: energy provisioning (**EP**) or server capacity provisioning (**CP**) in comparison with the joint optimization of doing both (**DCM**). The cost reductions of different optimization are shown in Fig. 8.

For the New York scenario in Fig. 8a, overall, we see that **EP**, **CP**, and **DCM** provide cost reductions of 16.3%, 7.3%, and 20.7%, respectively. However, note that during the day doing **EP** alone provides almost as much cost reduction as the joint optimization **DCM**. The reason is that during the high traffic hours in the day, solving **EP** to avoid higher grid prices provides a larger benefit than optimizing the energy consumption by server shutdown. The opposite is true during the night where **CP** is more critical than **EP**, since minimizing the energy consumption by shutting down idle servers yields more benefit.

For the San Jose scenario in Fig. 8b, overall, **EP**, **CP**, and **DCM** provide cost reductions of 6.1%, 19%, and 23.7%, respectively. Compared to the New York scenario, the reason why **EP** achieves so little cost reduction is that the grid power is cheaper and thus on-site generation is not that eco-



(a) Cost Reduction vs. look-ahead window size w

(b) Cost Reduction vs. percentage of on-site power production capacity

Figure 9: Variation of cost reduction with look-ahead and on-site capacity.

nomical. Meanwhile, **CP** performs closer to **DCM**, which is because the workload curve is highly skew (shown in Fig. 6b) and dynamic provisioning for the server capacity saves a lot of server idling cost as well as cooling and power conditioning cost.

In a nutshell, **EP** favors high grid power price while workload with less regular pattern makes **CP** more competitive.

6.4 Benefit of Looking Ahead

We evaluate the cost reduction benefit of increasing the look-ahead window. From Fig. 9a, we observe that while the performance of our online algorithms are already good when there is no look-ahead information, they quickly improve to the offline optimal when a small amount of look-ahead, *e.g.,* 6 hours, is available, indicating the value of short-term prediction of inputs. Note that while the competitive ratio analysis in Theorem 8 is for the worst case inputs, our online algorithms perform much closer to the offline optimal for realistic inputs.

6.5 How Much On-site Power Production is Enough

Thus far, in our experiments, we assumed that a hybrid data center had the ability to supply all its energy from on-site power generation ($N = 10$). However, an important question is how much investment should a data center operator make in installing on-site generator capacity to obtain largest cost reduction.

More specifically, we vary the number of on-site generators N from 0 to 10 and show the corresponding performances of our algorithms. Interestingly, in Fig. 9b, our results show that provisioning on-site generators to produce 80% of the peak power demand of the data center is sufficient to obtain all of the cost reduction benefits. Further, with just 60% on-site power generation capacity we can achieve 95% of the maximum cost reduction. The intuitive reason is that most of time the demands of the data center are significantly lower than their peaks.

7. RELATED WORK

Our study is among a series of work on dynamic provisioning in data centers and power systems [38, 22, 35].

In particular, for the capacity provisioning problem, [23] and [27] propose online algorithms with performance guarantee to reduce servers operating cost under convex and linear mixed integer optimization scenarios, respectively. Different from these two, our work designs online algorithm under non-linear mixed integer optimization scenario and we take into account the operating cost of servers as well as power conditioning and cooling systems. [24, 40] also model cooling systems, but focus on offline optimization of the operating cost.

Energy provisioning for power systems is characterized by unit-commitment problem (UC) [8, 31], including a mixed-integer programming approach [29] approach and a stochastic control approach [36]. All these approaches assume the demand (or its distribution) in the entire horizon is known *a priori*, thus they are applicable only when future input information can be predicted with certain level of accuracy. In contrast, in this paper we consider an online setting where the algorithms may utilize only information in the current time slot.

In addition to the difference of our work and existing works in the two problems (*i.e.*, capacity provisioning and energy provisioning), our work is also unique in that we jointly optimize both problems while existing works focus on only one of them.

8. CONCLUSIONS

Our work focuses on the cost minimization of data centers achieved by jointly optimizing *both* the supply of energy from on-site power generators and the grid, and the demand for energy from its deployed servers as well as power conditioning and cooling systems. We show that such an integrated approach is not only possible in next-generation data centers but also desirable for achieving significant cost reductions. Our offline optimal algorithm and our online algorithms with provably good competitive ratios provide key ideas on how to coordinate energy procurement and production with the energy consumption. Our empirical work answers several of the important questions relevant to data center operators focusing on minimizing their operating costs. We show that a hybrid (resp., on-grid) data center can achieve a cost reduction between 20.7% to 25.8% (resp., 7.3% to 12.3%) by employing our joint optimization framework. We also show that on-site power generation can provide an additional cost reduction of about 13%, and that most of the additional benefit is obtained by a partial on-site generation capacity of 60% of the peak power requirement of the data center.

This work can be extended in several directions. First, it is interesting to study how energy storage devices can be used to further reduce the data center operating cost. Second, another interesting direction is to generalize our analysis to take into account deferable workloads. Third, extension from homogeneous servers and generators to heterogeneous setting is also of great interest.

9. ACKNOWLEDGMENTS

The work described in this paper was partially supported by China National 973 projects (No. 2012CB315904 and 2013CB336700), several grants from the University Grants Committee of the Hong Kong Special Administrative Region, China (Area of Excellence Project No. AoE/E-02/08 and General Research Fund Project No. 411010 and 411011), and two gift grants from Microsoft and Cisco.

10. REFERENCES

[1] Akamai tech. http://www.akamai.com.
[2] Nationalgrid. https://www.nationalgridus.com/.
[3] Pacific gas and electric company. http://www.pge.com/nots/rates/tariffs/rateinfo.shtml.
[4] Tecogen. http://www.tecogen.com.

[5] The weather channal. http://www.weather.com/.

[6] Apple's onsite renewable energy, 2012. http://www.apple.com/environment/renewable-energy/.

[7] Distributed generation, 2012. http://www.bloomenergy.com/fuel-cell/distributed-generation/.

[8] C. Baldwin, K. Dale, and R. Dittrich. A study of the economic shutdown of generating units in daily dispatch. *IEEE Trans. Power Apparatus and Systems*, 1959.

[9] L. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 2007.

[10] A. Beloglazov, R. Buyya, Y. Lee, and A. Zomaya. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*, 2011.

[11] A. Borbely and J. Kreider. *Distributed generation: the power paradigm for the new millennium*. CRC Press, 2001.

[12] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

[13] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing energy and server resources in hosting centers. In *Proc. ACM SIGOPS*, 2001.

[14] A.J. Conejo, M.A. Plazas, R. Espinola, and A.B. Molina. Day-ahead electricity price forecasting using the wavelet transform and arima models. *Power Systems, IEEE Transactions on*, 2005.

[15] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959.

[16] R. Doyle, J. Chase, O. Asad, W. Jin, and A. Vahdat. Model-based resource provisioning in a web service utility. In *Proc. USITS*, 2003.

[17] K. Fehrenbacher. ebay to build huge bloom energy fuel cell farm at data center. 2012. http://gigaom.com/cleantech/ebay-to-build-huge-bloom-energy-fuel-cell-farm-at-data-center/.

[18] K. Fehrenbacher. Is it time for more off-grid options for data centers?. 2012. http://gigaom.com/cleantech/is-it-time-for-more-off-grid-options-for-data-centers/.

[19] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, and Alfons Kemper. Workload analysis and demand prediction of enterprise data center applications. In *Workload Characterization, 2007. IISWC 2007. IEEE 10th International Symposium on*, 2007.

[20] S. Kazarlis, A. Bakirtzis, and V. Petridis. A genetic algorithm solution to the unit commitment problem. *IEEE Trans. Power Systems*, 1996.

[21] J. Koomey. Growth in data center electricity use 2005 to 2010. *Analytics Press*, 2010.

[22] M. Lin, Z. Liu, A. Wierman, and L. Andrew. Online algorithms for geographical load balancing. In *Proc. IEEE IGCC*, 2012.

[23] M. Lin, A. Wierman, L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proc. IEEE INFOCOM*, 2011.

[24] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser. Renewable and cooling aware workload management for sustainable data centers. In *Proc. ACM SIGMETRICS*, 2012.

[25] J. Lowesohn. Apple's main data center to go fully renewable this year. 2012. http://news.cnet.com/8301-13579_3-57436553-37/apples-main-data-center-to-go-fully-renewable-this-year/.

[26] L. Lu, J. Tu, C. Chau, M. Chen, and X. Lin. Online energy generation scheduling for microgrids with intermittent energy sources and co-generation. In *Proc. ACM SIGMETRICS*, 2013.

[27] T. Lu and Chen M. Simple and effective dynamic provisioning for power-proportional data centers. In *Proc. IEEE CISS*, 2012.

[28] V. Mathew, R. Sitaraman, and P. Shenoy. Energy-aware load balancing in content delivery networks. In *Proc. IEEE INFOCOM*, 2012.

[29] J. Muckstadt and R. Wilson. An application of mixed-integer programming duality to scheduling thermal generating systems. *IEEE Trans. Power Apparatus and Systems*, 1968.

[30] E. Nygren, R. Sitaraman, and J. Sun. The Akamai Network: A platform for high-performance Internet applications. 2010.

[31] N. Padhy. Unit commitment-a bibliographical survey. *IEEE Trans. Power Systems*, 2004.

[32] D. Palasamudram, R. Sitaraman, B. Urgaonkar, and R. Urgaonkar. Using batteries to reduce the power costs of internet-scale distributed networks. In *Proc. ACM Symposium on Cloud Computing*, 2012.

[33] S. Pelley, D. Meisner, T. Wenisch, and J. VanGilder. Understanding and abstracting total data center power. In *Workshop on Energy-Efficient Design*, 2009.

[34] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath. Load balancing and unbalancing for power and performance in cluster-based systems. In *Workshop on compilers and operating systems for low power*, 2001.

[35] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *Proc. ACM SIGCOMM*, 2009.

[36] T. Shiina and J. Birge. Stochastic unit commitment problem. *International Trans. Operational Research*, 2004.

[37] M. Stadler, H. Aki, R. Lai, C. Marnay, and A. Siddiqui. Distributed energy resources on-site optimization for commercial buildings with electric and thermal storage technologies. *Lawrence Berkeley National Laboratory*, 2008.

[38] R. Stanojevic and R. Shorten. Distributed dynamic speed scaling. In *Proc. IEEE INFOCOM*, 2010.

[39] J. Tu, L. Lu, M. Chen, and R. Sitaraman. Dynamic provisioning in next-generation data centers with on-site power production. Technical report, Department of Information Engineering, CUHK, 2013. http://arxiv.org/abs/1303.6775.

[40] H. Xu, C. Feng, and B. Li. Temperature aware workload management in geo-distributed datacenters. In *Proc. ACM SIGMETRICS, extended abstract*, 2013.